Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Ujian Akhir Semester IF2211 Strategi Algoritma

Hari Rabu, Tanggal 5 Mei 2016

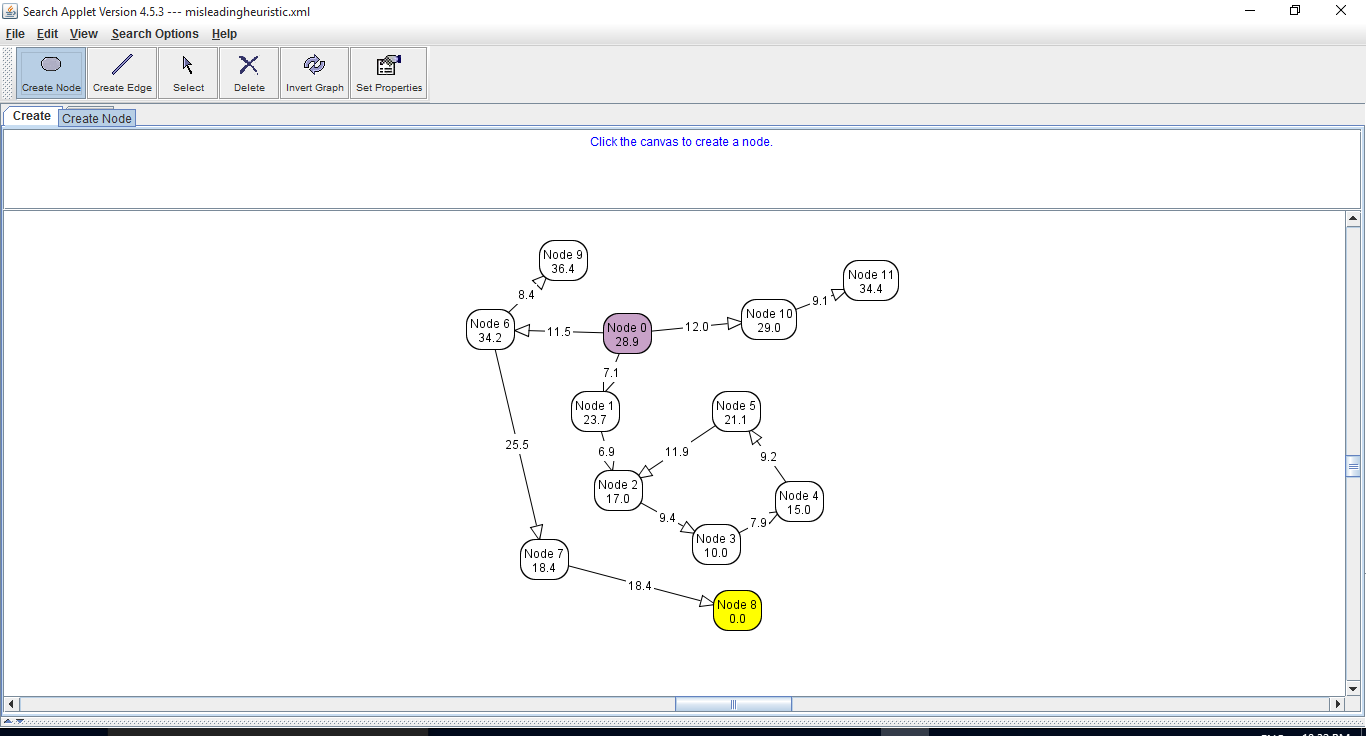
Waktu: 110 menit

Dosen: Rinaldi Munir & Nur Ulfa Maulidevi

*Berdoalah terlebih dahulu agar Anda sukses dalam ujian ini!*

**Bagian A (Soal dari Dr. Nur Ulfa Maulidevi)**

1. Pencarian jalur pada suatu graf dengan memanfaatkan algoritma A\* bertujuan untuk menghindari jalur yang ‘mahal’. Oleh karena itu nilai heuristik setiap simpul harus *admissible*. Dari Gambar 1 di bawah ini, tentukan apakah terdapat nilai heuristik yang tidak *admissible*, dan jelaskan dengan singkat. Nilai pada simpul adalah nilai heuristik, sedangkan nilai pada busur adalah biaya (*cost*) antar simpul. Pencarian jalur dimulai dari Node\_0 untuk menuju Node\_8.



Gambar 1. Graf berarah

**(Nilai = 10)**

1. Pada Gambar 2 berikut ini, terdapat graf yang menghubungkan simpul X ke simpul Z yang melalui beberapa simpul antara. Informasi mengenai jarak antar simpul dan nilai heuristik simpul (SLD dari simpul ke Z) sudah diberikan pada gambar tersebut.



Gambar 2. Graf lintasan X - Z

Lengkapilah tabel berikut untuk mencari jalur dari simpul X ke simpul Z dengan menggunakan beberapa pendekatan. Pencarian dihentikan ketika solusi pertama ditemukan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iterasi | Greedy Best First Search | | A Star | |
| Formula: f(n) = … {Isikan formula untuk mencari f(n) dengan Greedy Best First Search} | | Formula: f(n) = … {Isikan formula untuk mencari f(n) dengan A Star} | |
| Simpul – Ekspan | Simpul Hidup | Simpul - Ekspan | Simpul Hidup |
| 1 | X | AX f(AX) = … | X | AX f(AX) = … |
| f(…) = … | f(…) = … |  |
| Dst… |  |  |  |  |
| Hasil | Jalur = X - … - …  Jarak = …  Banyaknya iterasi hingga solusi pertama ditemukan = … | | Jalur = X - … - …  Jarak = …  Banyaknya iterasi hingga solusi pertama ditemukan = … | |

Catatan: Urutan anak adalah sesuai urutan abjad. Untuk tiap pendekatan, simpul yang sudah pernah di-**ekspan** dengan jalur yang sama tidak akan diperiksa lagi, dan tidak boleh membentuk sirkuit.

**(Nilai = 20)**

1. Terdapat persoalan *Knapsack* dengan kapasitas total bobot *Knapsack* 10. Terdapat 4 barang yang tersedia dengan bobot dan profit seperti pada Tabel 1. Selesaikan persoalan Knapsack tersebut dengan pendekatan *Branch and Bound*, dengan fungsi pembatas :

total\_profit\_barang\_terpilih + (sisa\_bobot\_yang\_masih\_memungkinkan \* nilai\_density\_terbaik\_berikutnya)

Tabel 1. Persoalan Knapsack

|  |  |  |  |
| --- | --- | --- | --- |
| Barang | Bobot | Profit | Profit/ Bobot  (Density) |
| 1 | 4 | 40 | 10 |
| 2 | 7 | 42 | 6 |
| 3 | 5 | 25 | 5 |
| 4 | 3 | 12 | 4 |

Gambarkan pohon pencarian, lengkap dengan keterangan setiap simpul merepresentasikan apa dan perhitungan lengkap untuk setiap simpul yang dibangkitkan pada pohon pencarian.

**(Nilai = 20)**

**Bagian B (Soal dari Dr. Rinaldi Munir)**

1. Carilah lintasan terpendek dari 1 ke 7 pada graf di bawah ini dengan algoritma Program Dinamis. Perlihatkn tabel perhitungannya (boleh pilih pendekatan maju atau mundur)

 **(Nilai = 20)**

1. (a) Berikan contoh sebuah *pattern* yang panjangnya 6 karakter dan text yang panjangnya 10 karakter sedemikian sehingga pencocokan untuk kasus terburuk pada algoritma KMP sama dengan pencocokan untuk kasus terburuk algoritma *Brute-Force*. Hitung jumlah perbandingan karakter yang terjadi.

(b) Diberikan sebuah pattern: EXAMPLE dan sebuah text: HERE IS A SIMPLE EXAMPLE. Jalankan algoritma Boyer-More untuk mencocokkan *pattern* dengan teks tersebut, perlihatkan proses pencocokannya *step by step*, lalu hitung jumlah perbandingan karakter yang terjadi.

(Catatan: gambarkan pattern dan teks sebagai larik, lalu tuliskan angka 1, 2, 3, … di atas *pattern* yang menyatakan jumlah perbandingan karakter yang terjadi) **(Nilai = 10 + 10)**

1. Diberikan tiga soal pilihan berganda di bawah ini, **pilihlah jawaban yang benar dan sertakan alasannya**:

(6.1). Dengan mengasumsikan P ≠ NP, maka dari jawaban berikut yang benar?   
(A) NP-complete = NP  
(B) NP-complete ∩ P = ∅  
(C) NP ∩ NP-complete = P   
(D) P = NP-complete

(6.2) Manakah dari pernyataan berikut yang BENAR? (1) Persoalan menentukan adakah lintasan terpendek di dalam graf tak-berarah adalah persoalan P. (2) Persoalan menentukan apakah terdapat lintasan terpendek di dalam graf tak-berarah adalah persoalan NP. (3) Jika persoalan A adalah NP-Complete, maka terdapat algoritma non-deterministik dalam waktu polinom untuk memecahkan A.

1. 1, 2, dan 3
2. 1 dan 3
3. 2 dan 3
4. 1 dan 2

(6.3) Sebuah persoalan di dalam NP adalah NP-complete jika   
(A) Ia dapat direduksi menjadi persoalan TSP dalam waktu polinom

(B) Persoalan TSP dapat direduksi menjadi persoalan tersebut dalam waktu polinom  
(C) Ia dapat direduksi menjadi sembarang persoalan lain di dalam NP dalam waktu polinom  
(D) Beberapa persoalan di dalam NP dapat direduksi menjadi persoalan tersebut dalam waktu polinom

**(Nilai: 12)**

1. Apa perkiraan nilai anda untuk mata kuliah ini? (A/AB/B/BC/C/D/E)  **(Nilai = 2)**